```
# The displaymenu function prints the menu. It's called from within the loop so when
# you've finished the operation you are on, you return to the menu - unless you've
chosen the exit option.
# the print("") statements just give a line break and keep things neat:

def displaymenu():
  print("Main Menu Options")
  print("-------------------------- ")
  print("1 to display records")
  print("2 to enter and add record")
  print("3 search record by ID")
  print("4 search by Author")
  print("5 search by Publisher")
  print("6 to update records")
  print("7 to delete records")
  print("8 to exit the programme")
  print(" ")

# In the select query function, notice the database connection starts with the 'with'
statement,
# moves on to the cursor statement, and then the SQL query. You can contain the
SQL inside the cursor.execute  statement - see the other queries.
# The select query makes a record set which is returned by the fetchall().
# We then loop through the record set displaying it using the 'for in result' loop.

def displayAllRecords():
  with sqlite3.connect('database.db') as mydb:
    cursor = mydb.cursor()
    sql = 'SELECT * FROM book';
    cursor.execute(sql)
    result = cursor.fetchall()
    for row in result:
      print ("BookID: " + str(row[0]) +", Author: " + str(row[1]) + ", Title:" +
str(row[2])+",Year: "+str(row[3]) +", Publisher: "+str(row[4]))


def enterAndInsertSingleRecord():
    #this uses variables
    enteredAuthor = input("Please enter Author's name >")
    enteredTitle = input("Please enter Book Title ")
    enteredYear = input("Please enter Year of publication ")
    enteredPublisher = input("Please enter Publisher ")
    with sqlite3.connect('database.db') as mydb:
     cursor = mydb.cursor()
     cursor.execute('insert INTO book (Null, author, title, year, publisher)
VALUES(?,?,?,?);' (enteredAuthor, enteredTitle, enteredYear, enteredPublisher))
     mydb.commit()
```

```
def deleteRecordByID():
 #prompt the user for the ID of the record to be deleted and put it in a variable
 idToDelete = int(input("Give the ID number of the record to delete."))
  with sqlite3.connect('database.db') as mydb:
  cursor = mydb.cursor()
  cursor.execute('DELETE FROM book WHERE bookID =?', idToDelete)
  mydb.commit()


def searchRecordByID():
    idToSearch = int(input("Give the ID of the record to search details of" )\
    with sqlite3.connect('database.db') as mydb:
  cursor = mydb.cursor()
  cursor.execute('SELECT * FROM book WHERE bookID =?', idToSearch)
  theRecord = cursor.fetchone()          #fetches a single record since ID is unique
  if theRecord.count == 0:
      print(" Record not found"
  else:
      print ("BookID: " + str(theRecord[0]) +", Author: " + theRecord(row[1]) + ", Title:" +
str(theRecord[2])+",Year: "+str(theRecord[3]) +", Publisher: "+str(therecord[4]))

#This next while loop has the call to display the menu and the if statements that
process the menu choice

loop = True
while loop:
  displaymenu()
  var_choice = int(input("Please make your choice using numbers 1-8."))
  print("")
  if var_choice == 1:
   displayAllRecords()
  elif var_choice == 2:
    enterAndInsertSingleRecord()
  elif var_choice == 3:
    searchRecordByID()
  elif var_choice == 4:
    searchByAuthor()                    #you have to write this module
  elif var_choice == 5:
     searchByPublisher()                #you have to write this module
  elif var_choice == 6:
      updaterecord()                    # you have to write this module
  elif var_choice == 7:
       deleteRecordByID()
  elif var_choice == 8:
   #this next statement turns the loop off
   loop = False
  elif var_choice =="":
   # the double quotes together are a null value; if you do not enter a value
   print("Please enter a number. ")
  else:                          # invalid number entered (out of range 1-8)
    print("Invalid input.")
```

```
# The insert query has two variables which are collected in an array  -  these
variables are represented by ? marks in the SQL statement.
# The question marks are parameters that you load values into from the array
variables outside the SQL.
# it's very important that you 'load' the variables in the correct order - note the
var_insert array [].
# Notice also the counter which increments with each run of the while loop.
# Note - the  configuration of the cursor.execute and the fact that the SQL finishes
with the db.comit()

def insert():
  var_choice2 = int(input("Select the number of records to input "))
  var_count = 1
  while var_choice > var_count:
    var_menuitem = str(input("Input menu item. "))
    var_cost = str(input("Input cost(no currency symbols). "))
    var_insert = []
    var_insert.append(var_menuitem)
    var_insert.append(var_cost)

    with sqlite3.connect('db_food') as db:
      cursor = db.cursor()
      cursor.execute('insert INTO tbl_menu (fld_menu_item, fld_cost)VALUES(?,?);',
var_insert)
      db.commit()
      var_count += 1


#The update query has three values appended to the var_update array, again they
need to be appended in the correct order.

def update():
  var_updt = str(input("Give the ID number of the record to update. "))
  var_updt_menu_item = str(input("Give the new wording of Menu Item Field. "))
  var_updt_cost= str(input("Give the new cost of the Cost Field. "))
  var_update = []
  var_update.append(var_updt_menu_item)
  var_update.append(var_updt_cost)
  var_update.append(var_updt)
  with sqlite3.connect('db_food') as db:
    cursor = db.cursor()
    cursor.execute('UPDATE tbl_menu SET fld_menu_item = ?, fld_cost = ? WHERE
fld_foodID =?', var_update)
    db.commit()
```

Create a test plan and then test. Remember any validations tend to involve several test;

Add validation rules for adding a new record and put appropriate tests in the test plan

Test no	Description of test	Input data	Expected outcome
1	Valid menu option	5	
2	Invalid menu option low	0	
3	Lower boundary	1	
4	Upper	8	
5	Invalid upper	10	
6	Add Record	Author = "JNN" Title = "Single Table Database" Year = " 2025" Publisher = "HSFC"	DisplayALL before should display records without but after adding, DisplayAll should now show the new record
7	Year validation?		
8	Blank Author?		
9	Blank Title?		
10	Blank Publisher?		
11			
12	Search by ID		
13	Delete by ID		